CLAIMS

What is claimed is:

1. A method of scheduling in a mixed workload environment on a computing system having a CPU resource and a permanent storage resource, the computing system servicing requests from one or more clients, comprising:

executing a current process on the CPU resource and the storage resource, the current process having been dispatched to service a current client request;

performing a contention check while executing the current process to determine whether a new client request has a transaction priority that is greater than the transaction priority of the current client request;

if the transaction priority of the new client request is greater than that of the current request, dispatching a process to service the new client request;

if the transaction priority of the new request is not greater than that of the current request, determining whether the transaction priority of the current request is less than a predetermined threshold priority;

if the transaction priority of the current client request is lower than the predetermined threshold priority and there is higher priority I/O activity present on the storage resource:

delaying the servicing of the current client request and forgoing the servicing of any read aheads for the current client request; and

dispatching a process to service the highest priority client request that is available for service; and

if the transaction priority of the current client request is greater than the predetermined threshold or the priority of the current client request is lower than the predetermined threshold and there is no higher priority I/O activity present on the storage resource:

determining whether the current client request requires any read aheads;

dispatching one or more helper processes to service any required read aheads; and

returning to the current process to service the current client request.

2. A method of scheduling in a mixed workload environment as recited in claim 1,

wherein a maximum priority in the system is 255 and a minimum priority is 1; and

wherein the threshold priority is 151.

3. A method of scheduling in a mixed workload environment as recited in claim 1, wherein the step of delaying the servicing of the current client request includes

delaying the servicing of the current client request by an amount of time that depends on the transaction priority of the current client request, higher priority requests being delayed less than lower priority requests, and the amount of the delay being the sum of a fixed delay and a priority dependent delay.

4. A method of scheduling in a mixed workload environment as recited in claim 3,

wherein a maximum transaction priority in the system is 255 and a minimum priority is 1 and the threshold priority is 151;

wherein the fixed delay is about 0.2 seconds; and

wherein the priority dependent delay is the product of a constant and the difference between the threshold priority and the priority of the current client request.

5. A method of scheduling in a mixed workload environment as recited in claim 4, wherein the constant is approximately 0.02.

6. A method of scheduling in a mixed workload environment as recited in claim 1, wherein the step of delaying the servicing of the current client request includes delaying the servicing by a fixed delay.

7. A method of scheduling in a mixed workload environment as recited in claim 6, wherein the fixed delay is approximately 10 milliseconds.

8. A method of scheduling in a mixed workload environment as recited in claim 1, wherein the step of performing a contention check occurs once every time a physical block is transferred from the storage resource.